# TRADEOFFS IN DESIGNING ORGANIZATIONS: IMPLICATIONS FOR NEW FORMS OF HUMAN ORGANIZATIONS AND COMPUTER SYSTEMS

THOMAS W. MALONE
STEPHEN A. SMITH

MARCH 1984

CISR WP #112
SLOAN WP #1541-84



CENTER FOR INFORMATION SYSTEMS RESEARCH
SLOAN SCHOOL OF MANAGEMENT
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

## Abstract

In this paper, we develop a model that can be applied to a wide range of problems in organization theory and computer science including: (1) explaining historical changes in the structure of American business organizations, (2) predicting changes in the structure of human organizations that may result from the widespread use of computers, and (3) analyzing the advantages and disadvantages of decentralized task scheduling in computer networks.

To analyze these diverse problems, we argue that one of the fundamental problems that must be solved by all organizations (including both human organizations and computer systems) is the task assignment problem, and we show how different organizational structures can be regarded as alternative solutions to this problem. Viewing organizational structures in this way provides insights into fundamental tradeoffs in designing organizations such as the tradeoffs between flexibility and efficiency. Using queuing theory and some simple assumptions about assignment methods, we are able to derive analytic justifications for qualitative statements of these tradeoffs. Finally, we suggest how this analysis can be applied to a variety of organizational design issues including those listed above.

# Tradeoffs in Designing Organizations: Implications for New Forms of Human Organizations and Computer Systems

Thomas W. Malone and Stephen A. Smith

A number of observers have predicted that the dramatic increases in power and affordability of various forms of information technology will have widespread ramifications for the structure of modern organizations (e.g., Huber, in press; Naisbitt, 1982; Scott Morton & Rockart, 1983; Strassman, 1980). This paper presents a principled basis for evaluating these predictions and for advising managers and others about their new organizational choices. Since the primary factors changing in this situation involve information processing, the model we will present emphasizes these factors. This abstract model of information processing has implications for designing, not only human organizations, but also "organizations" of computer processors and organizations that combine people and computers. For example, our model helps analyze the following situations:

> (1) In the past century, the dominant organizational structures in American businesses changed from numerous small firms to large functionally-organized hierarchies (roughly 1850 - 1910) and then to large multidivisional hierarchies segmented by product line (roughly 1920 - 1960, see Chandler, 1962, 1977; Williamson, 1981b). How can we explain these changes and how can we predict future changes that may result from the widespread use of computers?
>
> (2) Computer systems are increasingly being designed to take advantage of many processors that operate in parallel and are sometimes geographically separated (e.g., Siewiorek, Bell, & Newell, 1982; Jones & Schwarz, 1980; Lorin, 1979). These systems present a number of new choices for methods of coordinating activities in different processors. How can we evaluate these choices?

In order to analyze these diverse situations, we first define a set of fundamental organizational structures that are used in both human organizations and computer systems. In each case, we show how these structures can be seen as alternative solutions to the problem of assigning tasks to processors. Then using straightforward mathematical analysis and results from queueing theory, we derive qualitative statements about trade-offs between factors such as efficiency and flexibility in the alternative structures. Finally, we suggest how these results can be used to analyze problems in both computer science and organization theory, including the problems described above.

One important difference between our analysis and most previous analyses of organizational structure (e.g., Lawrence & Lorsch, 1967; Woodward, 1965) is that we explicitly include not only

features of the production technology (e.g., economies of scale) and of the environment (e.g., importance of flexibility), but also features of the coordination technology itself (e.g., communication costs). Several previous economic analyses have also included communications costs (e.g., Williamson, 1975, 1979, 1980, 1981a, 1981b; Coase, 1937; Jonscher, 1981), and our analysis complements these in that it includes the load leveling and failure tolerance characteristics of organizational structures as well as certain aspects of their transaction costs.

*Definition of organization*

In order to analyze organizations at this abstract level, we define an *organization* as consisting of:

(a) a group of *agents* (either people or machines),
(b) a set of *activities* performed by the agents,
(c) a set of *connections* among the agents, and
(d) a set of *goals* or evaluation criteria by which the combined activities of the agents are evaluated.

To *organize*, then, is to

(1) establish (either explicitly or implicitly) the goals of the organization[1],
(2) segment the goals into separate activities, and
(3) assign the activities to agents in such a way that the overall goals are achieved.

In this paper, we focus on the third problem: assigning activities to agents. We will often refer to activities as "tasks", agents as "processors," and to this problem as the "task assignment problem."

Note that this definition of "organization" is much broader than that of many authors (e.g., see Scott, 1981) since it includes any group of agents whose activities are coordinated to achieve some overall goals. Thus, an "organization" in our sense might consist of the employees of a corporation or the members of a voluntary association, or it might be all the buyers and sellers in the market for a particular set of products. The buyers and sellers in a market are regarded as an "organization" because, even though they may have many different individual goals, their joint actions result in achieving the goal of producing and allocating the products being sold. In most cases, our models are relatively insensitive to where the legal boundaries of organizations are drawn, since we are concerned primarily with the information processing necessary to assign tasks.

*The task assignment problem*

March and Simon (1958, p. 22) note that one could begin a formal theory of departmentalization by

regarding the total set of tasks necessary to achieve the organizational purpose as given in advance. Then the problem of assigning tasks to people can be formulated mathematically as one of minimizing the total costs, subject to certain constraints such as no person having more than 8 hours of work to do per day. One of the most important problems with this approach, as March and Simon point out, is that all tasks are regarded as fixed in advance. In fact, many of the tasks performed in organizations are not known in advance and must be assigned as they arise. It is precisely this dynamic assignment aspect of the coordination problem that we address in this paper. We can formulate the problem in general terms as follows:

Given:

(1) a set of tasks generated over time with different requirements, and
(2) a set of processors with different capabilities.

Determine:

how to assign the tasks to the processors in order to achieve some overall objectives.

Examples of this general problem include: (1) designing, manufacturing, and marketing products, where the processors are people or machines with the special capabilities required, and (2) processing multi-step computer jobs using different processors on a network of computers.

*Scope of the model*

In general, our model focuses on certain information processing aspects of coordination and omits many other factors that are important in human organizations (such as power relationships, opportunism, and individual motivation). In addition to helping us understand the aspects we analyzed, this approach can also provide a "baseline" for comparison that helps us understand other aspects, as well. For example, we describe instances below where our model predicts that either of two organizational forms should be equally desirable, but where one of the two, in fact, occurs more often. In these cases, other factors (such as opportunism) appear to be necessary to explain the observed results.

In particular, we are explicitly *not* attempting to analyze how goals are established or how they change (e.g., Cyert & March, 1963), how goals are subdivided into separate activities, or how the results of separate activities are reintegrated (e.g., Lawrence & Lorsch, 1967). Neither do we attempt to model the power and authority aspects of decision making (e.g., Pfeffer, 1980). Our analysis of decision-making is concerned only with where decisions are made and how they are communicated, not with how they are enforced. Finally, our analysis is based on "pure" forms of

different organizational structures. Almost all real organizations are mixtures of these pure forms (e.g., see Ansoff & Brandenburg, 1971).

In spite of these simplifications from actual organizational realities, however, our analysis seems to account for a large range of the important tradeoffs between alternative organizational forms in both human organizations and computer systems. In fact, we believe that our results apply in practice to a much wider range of situations than are captured by the simple assumptions of the formal model.

## ALTERNATIVE ORGANIZATIONAL FORMS

Figure 1 shows several simple organizational forms that can be used to solve the task assignment problem. These simple forms serve as building blocks for much more complex organizations. The names for these organizational forms are taken from the vocabulary of human organizations, but as we will show below, there are also computer organizations that are analogous to each form.

In each form, there are several different processor types, each of which can perform a certain kind of task. The organizational forms differ in the following ways: (1) whether or not tasks are shared among processors of the same type, (2) whether the processor scale is large or small, and (3) whether decision making about task assignment is centralized or decentralized.

In order to compare the different forms, we assume that they are identical in terms of the following: (1) the "products" that must be produced to achieve the organizational goals, (2) the tasks that must be performed to produce these products, (3) the total processing power available for performing each kind of task, (4) the cost of operating the processors, and (5) the difficulty of deciding what tasks need to be done and what kind of processor can do them.

We will compare the different organizational forms in terms of their production costs, their coordination costs, and their "vulnerability costs". The assumptions we describe in this section will allow us to measure these factors in the following terms: (1) production costs in terms of the delay in processing tasks, (2) coordination costs in terms of the minimum number of communication instances, or "messages", necessary to assign tasks to processors, and (3) vulnerability costs in terms of the costs of unexpected changes such as component failures.

**Product hierarchy**

*Human organizations.* In a product hierarchy, there is a separate division for each product or major product line. Each division has a "product manager" and its own separate departments for different kinds of functions such as marketing, manufacturing, and engineering. In this form, the "executive office" may set long-range strategic directions, but it is not ordinarily involved in the operational assignment of tasks to processors. The lack of connection with the executive office for scheduling purposes is indicated by dotted lines in Figure 1. (This form is sometimes called the "multi-divisional" form (Chandler, 1962) or the "M-form" (Williamson, 1975).)

*Computer systems.* In computer systems, one analog of a product hierarchy is a group of separate personal computers, not connected by a network. Each personal computer can be considered a separate "division" with its own dedicated "departments" for performing different kinds of functions. In this case, however, the functions to be performed are not tasks like marketing, manufacturing, and engineering. Instead they are tasks like printing information on the printer, storing and retrieving information from the disk drive, and displaying information on the screen. The "product manager" that assigns tasks to all the different departments is the processing unit of the microcomputer. The processing unit of the microcomputer can also be considered to contain one of the functional departments--the department that performs the arithmetic and logical computations.

*Task assignment method.* The solution to the task assignment problem that is implied by this organizational form is simple: Whenever a task of a certain type needs to be done, the product manager assigns the task to the department that specializes in that type of task. In the "pure" form of a product hierarchy, there is only one department (or one processor) for each type of task, so the assignment decision is trivial. Of course, deciding which kind of department or person is best suited to perform a particular task may not be a trivial matter at all, but as noted above, we assume that it is equally difficult in all the organizational forms. Once the task has been classified as one needing a particular type of processor, the decision about which processor of that type it should be assigned to is trivial in this case, since there is only one processor of that type in the division.

We will assume that, in the simplest case, one message is required to assign a task to a processor and one message is required for the processor to return the result. Returning the result may, in some cases, consist of simply notifying the product manager that the task has been completed.

When a processor fails in a product hierarchy, the product division in which the failure occurs is

disrupted, but the other divisions are not affected.

Functional hierarchy

*Human organizations.* In a functional hierarchy, the processing needs for all products are pooled in functional departments. Each functional department has a "functional manager" and one or more processors of the same kind (e.g., a marketing department with a number of marketing specialists or a manufacturing department with a number of interchangeable production lines). As Figure 1 indicates, the functional departments may be composed of a number of small scale processors or a single large scale processor (in order to take advantage of economies of scale). In a functional hierarchy, the "executive office" coordinates the functional processing for all the different products by dispatching tasks to the appropriate functional departments. (This form is sometimes called the "unitary" form or "U-form" (Williamson, 1975).)

*Computer systems.* In computer systems, one example of a functional hierarchy is a time-sharing system with centralized (and often large-scale) "departments" for arithmetic and logical processing, printing, file storage, and so forth. As before, the "executive" function is physically merged with one of the functional departments--the arithmetic and logical processor. In many time-sharing systems, this central processor acts as the functional manager for other departments as well (e.g., scheduling jobs in the printer queue).

*Task assignment method.* The task assignment method implied by the "pure" form of this organizational structure is somewhat more complicated than for the product hierarchy, because an extra layer of management is involved: Whenever a task of a certain type needs to be done, the executive office delegates it to the functional manager of the appropriate type who, in turn, assigns it to one of the processors in that department. In order to make this assignment intelligently, the functional manager needs to keep track of not only the priorities of the tasks, but also the loads and capabilities of the processors in the department.

We assume that, in the simplest case, four messages are required for each task: one to delegate it to a functional department, one to assign it to a processor, one to return the result to the functional manager, and one to notify the executive office.

When a small scale processor fails in a functional hierarchy, the tasks it would have performed are delayed until they can be reassigned to another processor. When a large scale processor fails or when the functional manager fails, the entire organization may be disrupted.

**Decentralized (competitive) market**

*Human organizations.* In a decentralized market, "clients" search in the marketplace for acceptable "contractors" to perform the tasks they need to have done. Clients may have some prior knowledge of the capabilities of different kinds of processors, but in general, they must communicate with a number of potential contractors in order to determine the specific capabilities and current availability of the contractors. In some cases, the clients would be called "customers" and the tasks performed by the contractors would be supplying products rather than performing services. As noted above, our model is relatively insensitive to where legal boundaries between organizations are drawn. For example, in the applications below, we will suggest that some of the "lateral" information processing that occurs among different parts of hierarchical organizations can be modeled as a decentralized market.

*Computer systems.* There are several examples of computer systems organized as decentralized markets (Malone, Fikes, and Howard, 1983; Smith and Davis, 1981; Farber and Larson, 1972). For example, the Enterprise system (Malone, Fikes, and Howard, 1983) is a decentralized scheduler that allows personal computers connected by a network to share tasks in a way that assigns tasks to the "best" available processor at any time. Processors with tasks to be done are clients and other unused processors on the network are contractors. Clients send out "requests for bids" for tasks to be done and potential contractors respond with "bids" indicating their availability or cost for performing the tasks. The clients then select a bidder and send the task to the winning bidder.

*Task assignment method.* In a decentralized market, clients send announcements of tasks to some fraction of the contractors of the appropriate type and then receive bids from some of those contractors. In addition to these bidding messages, we also assume that each task requires one message to assign it to a processor and one message to return the result.

When a processor in a decentralized market fails, the tasks it would have performed are delayed until they can be reassigned to another processor.

With large numbers of clients and contractors, the "search costs" implied by all the announcement and bid messages may be considerable. As the next organizational form shows, these costs can be substantially reduced by using brokers or other centralized coordinators.

### Centralized (monopolistic) market

*Human organizations.* In a completely centralized market, clients do not need to search for contractors because there is only one source for each type of processor. For our purposes, the scheduling results will be the same whether this central source is (1) an exclusive broker who matches clients and contractors for a commission, (2) a sole-source contractor who manages a number of "captive" subcontractors, or (3) a firm that includes all the subsidiary processors. It is also possible (though not shown in the figure) for one central scheduler to manage several types of functional processors.

*Computer systems.* It is easy to imagine the Enterprise system (Malone, Fikes & Howard, 1983), for example, with one processor on the network serving as a single centralized scheduling node. Instead of broadcasting announcements of tasks to be done to all available contractors, client machines would simply send their requests to the scheduling node. The scheduling node would keep track of the availability of all the processors on the network and send the task to the best processor when it became available.

*Task assignment method.* From a task assignment point of view, a centralized market is identical to a functional hierarchy. Both have a single central scheduler for each type of task, both require the same number of messages for assignment (four per task in the simplest case), both have the same amount of load sharing among processors, and both have the same responses to component failures. Both also have the possibility of replacing several small processors with one large one to take advantage of whatever economies of scale are present in the underlying technology. Thus one of the insights provided by this analysis is that these two forms, which would often be considered very different, are identical in terms of the information processing variables we are considering here.

### Other organizational forms

As mentioned above, these four "pure" organizational forms serve as building blocks for the much more complex organizations we observe. For example as Figure 2 shows, a "matrix" organization is a hybrid form in which a functional hierarchy is augmented by separate product managers for each product who have direct links to specialized processors in each functional division. From a task scheduling point of view, this might imply that specialized processors give priority to tasks from the product manager to which they are linked but that all specialized processors in a department are available to help with each others' overflow tasks.

Other examples of composite organizational forms include (1) product hierarchies in which each product division is organized as a small functional hierarchy with multiple small scale processors in

each department, (2) decentralized markets in which contractors are internally organized as functional hierarchies, (3) organizations in which a formal product hierarchy is supplemented by informal communications and load-sharing patterns that resemble a decentralized market, and (4) regulated markets in which a hierarchical structure (for example, a functional hierarchy) is superimposed on a decentralized market.

### Summary of assumptions

Table 1 summarizes the assumptions we have made about the different organizational forms. The chart combines centralized markets and functional hierarchies on the same lines, since in terms of the factors we consider here, they are the same. As noted above, these forms are equivalent in terms of the tasks they perform and the processing power they have. They differ only in the way task performance is coordinated. In the next section we examine what differences these coordination mechanisms make.

## TRADEOFFS AMONG ORGANIZATIONAL FORMS

Table 2 shows the tradeoffs among the different organizational forms on several evaluation criteria. This table is a summary of the major results in the entire paper. The remaining sections of the paper present justifications for the results in Table 2 and show how they can be applied.

The evaluation criteria in Table 2 are divided into two groups: *efficiency* and *flexibility*. Efficiency is broken into two parts: (1) *production costs* (represented here as average delay in performing tasks) and (2) *coordination costs* (represented here as the cost of transmitting and processing the messages used to construct schedules). As noted above, we assume that the operating costs of the production technology are the same in all organizational forms. Therefore, a lower average delay in performing tasks means lower average production costs.

Even though flexibility is often thought of as the opposite of efficiency, it is clear that flexibility essentially means long-run efficiency in changing environments.[2] In other words, the flexibility/efficiency tradeoff is a tradeoff between short-run and long-run efficiency. We distinguish two kinds of measures for long-run flexibility: (1) *vulnerability*, the cost of unavoidable degradation in performance which a system suffers when the situation changes (represented here as the average cost of component failure), and (2) *adaptability*, the cost of adapting to a changed environment in order to achieve the same level of performance as before the change (represented here as the coordination costs of rescheduling). Note that coordination costs affect both short-run

**Table 1**
**Assumptions about Alternative Organizational Forms**

| *Organizational form* | *Production Costs* | | *Coordination Costs* | | *Vulnerability Costs* | |
|---|---|---|---|---|---|---|
| | *Processors shared among products* | *Processor scale* | *Centralization of decision-making* | *Minimum no. of messages to assign task to best processor* | *Result of processor failure* | *Result of scheduler failure* |
| Product hierarchy | No | Small | No | 2 | division disrupted | --- |
| Decentralized market | Yes | Small | No | 2m+2 * | task reassigned | --- |
| Centralized market/ Functional hierarchy | Yes | Small | Yes | 4 | task reassigned | entire org. disrupted |
| Centralized market/ Functional hierarchy (Large scale) | Yes | Large | Yes | 4 | entire org. disrupted | entire org. disrupted |

* Note: m is the number of processors in the market.

Table 2
Tradeoffs Among Alternative Organizational Forms

| *Organizational form* | *Evaluation Criteria* | | |
|---|---|---|---|
| | *Efficiency* | | *Flexibility* |
| | *Production Costs* (Average delay) | *Coordination Costs* (Message processing costs) | *Vulnerability Costs* (Average cost of component failure) |
| Product hierarchy | H | L | M |
| Decentralized market | M | H | L |
| Centralized market/ Functional hierarchy | M | M | M' |
| Centralized market/ Functional hierarchy (Large scale) | L | M | H |

Note:

L = Low costs ("good")
M = Medium costs
H = High costs ("bad")

efficiency and also, when rescheduling is necessary, long-run flexibility.

All the evaluation criteria shown in the chart are represented as costs, so in every column low is "good" and high is "bad." Primes are used to indicate indeterminate comparisons. For example, M' is less than H and greater than L, but it may be either greater or less than M.

First of all, the table shows very clearly that design choices often involve a complex trade-off among several dimensions. In some cases, detailed estimates of the parameter values are necessary to predict which forms are preferable. In many cases, however, the qualitative form of the tradeoffs shown here enables us to make inferences about situations where only one factor changes and all others remain constant, or where several factors all change in the same direction.

**Summary of previous organizational design principles**

The qualitative comparisons shown in the table provide a concise summary of many of the generalizations about organization design that have been made by previous theorists (e.g., Galbraith, 1977; March & Simon, 1958; Gulick and Urwick, 1937). For example, March and Simon (1958, p. 29) summarize the problem of departmentalization as centering on a tradeoff between self-containment and skill specialization: "[Functional] departmentalization generally takes greater advantage of the potentialities for economy through specialization than does [product] departmentalization; [product] departmentalization leads to greater self-containment and lower coordination costs. . ."[3] Table 2 reflects this tradeoff with the "economies of specialization" in functional hierarchies being represented as lower production costs, and the advantages of self-containment in product hierarchies being represented as lower coordination costs.

Galbraith (1977), extends this view by pointing out that the advantages of coordination can be obtained by either investment in a vertical information system (as in a functional hierarchy in Table 2), or by the creation of lateral relations (as in a decentralized market in Table 2).

Our model also reflects the vulnerability costs implied by observations such as Gulick's (Gulick & Urwick, 1937, p. 24): "A failure in one [product division] is limited in its effect to that service. . . . A failure in one [function] affects the whole enterprise, and a failure to co-ordinate one [functional] division, may destroy the effectiveness of all the work that is being done."[3] This difference is shown in Table 2 where product hierarchies have a lower vulnerability cost than large scale functional hierarchies. Interestingly, this difference does not necessarily hold if the processors are the same scale in both cases, since the cost of disrupting an entire product division might be greater than the cost of simply rescheduling a task from one failed processor to another one in a functional

hierarchy.

In all these cases, our model not only summarizes previous results but also places them in a more comprehensive framework. For example, March and Simon and Galbraith did not give much consideration to markets as alternatives to product hierarchies or functional hierarchies, and Gulick did not seem to appreciate the difference that processor scale might make in the vulnerability arguments he presented.

### Size of the economy

The tradeoffs shown above in Table 2 assume that the size of the "economy" being modeled is fixed, that is, that the number of processors and the total number of managers generating tasks are all constant. As the size of the economy increases, the relative rankings of the alternative organizational forms do not change on any of the evaluation criteria. However, the values change much faster for some organizational forms and criteria than for others. Thus simply changing the size of the economy, even without changing any other parameter values, may change the relative importance of different criteria and therefore change the "optimal" organizational form. The relative rates of change for the different criteria are summarized in Table 3 and justified in the next section and the appendix. A zero in Table 3 indicates that no change occurs with size. One or more pluses or minuses after a letter indicate the relative rates with which the criteria change as size increases.

## JUSTIFICATIONS FOR ORGANIZATIONAL FORM COMPARISONS

In the appendix, we present formal justifications for the ordinal rankings defined by the qualitative comparisons in Tables 2 and 3. In this section, we give brief intuitive explanations of the comparisons. The intuitive justifications in this section should provide an adequate basis for understanding the remainder of the paper.

### Assumptions

In addition to the assumptions summarized in Table 1, the following additional assumptions are made:

1. Tasks are randomly generated.
2. Processing each task takes a random amount of time.
3. Coordination costs are proportional to the number of messages sent between agents to

Table 3
Changes in Evaluation Criteria as Size of the Economy Increases

| *Organizational form* | *Evaluation Criteria* | | |
|---|---|---|---|
| | *Efficiency* | | *Flexibility* |
| | *Production Costs* (Average delay) | *Coordination Costs* (Message processing costs) | *Vulnerability Costs* (Average cost of component failure) |
| Product hierarchy | 0 | 0 | ++ |
| Decentralized market | - | + | + |
| Centralized market/ Functional hierarchy | - | 0 | ++ |
| Centralized market/ Functional hierarchy (Large scale) | - | 0 | +++ |

Note:

0 = No change
+ = Increase
- = Decrease

assign tasks.

4. Large processors are m times faster than small ones and cost m times as much. (In other words, the simple model assumes that there are no economies of scale. This assumption is relaxed in some of the applications.)

5. Large processors fail at least as often as small ones.

6. Scheduling managers sometimes fail (i.e., with probability greater than 0).

7. The cost of delaying a job in order to reassign it is less than the cost of disrupting an entire division or organization.

8. The cost of disrupting an entire organization is at least m times as much as the cost of disrupting a division.

9. If the economy increases in size, both the processing load and the total processing power increase at the same rate.

**Production costs**

The product hierarchy has the highest average delay in processing tasks because it uses slow processors that are not shared. The decentralized market, centralized market, and functional hierarchy all have a somewhat lower average delay time because they are able to take advantage of the "load leveling" that occurs when tasks are shared among a number of similar processors. For example, processors that would otherwise be idle can take on "overflow" tasks from busy processors thus reducing the overall average delay. Finally the large scale versions of a centralized market and a functional hierarchy have the least average delay time because they not only have the load leveling advantages of pooled processors, but they also are always working with the full processing speed of a large machine, even at times when some of the small machines would be idle.

**Coordination costs**

The product hierarchy requires the least number of messages for task assignment since each task is simply sent to the processor of the appropriate type in the division in which the task originates. The centralized market and functional hierarchy require more scheduling messages since tasks must be sent to a centralized scheduling manager (e.g., a functional manager or a broker) before being sent to the proper processor. The decentralized market requires the most messages of all since assigning each task requires sending "requests for bids" to a number of possible processors of the appropriate type and then receiving bids in return.

**Vulnerability costs**

The decentralized market is the least vulnerable to component failure since if one processor fails,

the task is only delayed until it can be transferred to another processor. The centralized market and functional hierarchy are somewhat more vulnerable since, not only can tasks be delayed by the failure of individual processors, but also the entire system will fail if the centralized scheduling manager fails. The product hierarchy is also more vulnerable than the decentralized market because when a processor fails, tasks cannot be easily transferred to another similar processor. Finally, the centralized market and functional hierarchy with large scale processors are the most vulnerable of all because large scale processors fail at least as often as small ones, and when they do, all the tasks in the system fail at once.

### Size of the economy

As the number of processors in the economy increases, the average delay time for the product hierarchy does not change at all, but the load leveling and processing speed advantages of the other forms become more and more important. Similarly, the number of messages required for scheduling does not increase at all with size for the product hierarchy, functional hierarchy, or centralized market, but the number of messages required to get bids from all the eligible processors in a decentralized market increases in proportion to the number of processors in the system. Finally, as the number of processors increases, the failure rate per unit time increases in all organizational forms, but it increases most in the large scale centralized market and functional hierarchy and least in the decentralized market.

## APPLICATIONS

In this section, we will suggest how the analysis just presented can be applied to a wide variety of organizational design issues. Several of the applications described here are not fully developed; they are included to suggest directions for future development.

### Computer systems

#### *Decentralized scheduling for computer networks*

As mentioned earlier, several computer systems (Malone, Fikes, and Howard, 1983; Smith and Davis, 1981; Farber and Larson, 1972) use decentralized task scheduling techniques based on market metaphors. We are now in a position to evaluate some of the advantages and disadvantages of such decentralized systems in comparison to centralized systems that have a single scheduling manager.

As shown in Table 2, the primary advantage of the decentralized system is its high reliability (low vulnerability costs), and its primary disadvantage is the number of messages that must be transmitted back and forth to construct schedules (high coordination costs). Which of these factors is most important in a given situation depends on the system load, the cost of sending bidding messages, the reliability of the machines involved, and the costs of scheduler failure.

In particular, we can derive exact comparisons between the centralized and decentralized markets by measuring all the costs shown in Table 6 in terms of time lost by users. First, we let

D = the expected total time lost (in seconds per user per day)

$p_{SCHED}$ = the probability of scheduler failure,

f = the expected time lost if the scheduler fails (in seconds per user),

t = the expected number of tasks per user per day scheduled by the system, and

d = the expected additional delay per task required to wait for and process bids, beyond the delay that would be required by a centralized scheduler (in seconds).

Then it is a straightforward matter to show that the difference in cost between the systems is

$$D_{DM} - D_{CM} = td - p_{SCHED}f.$$

For illustration, we use the following very rough estimates of these values for the environment in which the prototype Enterprise system (Malone, Fikes, & Howard, 1983) was implemented:

$$D_{DM} - D_{CM} = (20)(1) - (.03)(900) = -7.$$

In other words, according to these very rough estimates, a decentralized scheduling system has a small advantage (7 seconds per user per day) over a centralized one in this environment. Large changes in at least one of the factors would be necessary for either approach to have a decisive advantage.

*Trends in computer system architectures*

Another issue our model can help illuminate is the effect on computer system architectures of changing costs of the underlying technologies (c.f., Frazier, 1979; Lorin, 1979). For example, one of the most obvious changes in computer system architectures in recent years has been the explosive growth in the use of personal computers relative to time-shared mainframe computers. In terms of our model, computational power is increasingly being supplied by product hierarchies (personal

computers) rather than large scale functional hierarchies (mainframes). We can explain this result by observing that since the unit costs of computation are decreasing dramatically, minimizing computational cost is no longer as important as it once was. Therefore, according to Table 2, the shift toward product hierarchies should occur because product hierarchies save coordination costs and the extra computational power they require is no longer as expensive. The kinds of coordination costs saved in this case might include: (1) the costs of transmitting data between the user's terminal and a distant processor, (2) the operating system overhead required when the processor is shared by a number of users, and (3) the administrative overhead of dealing with a computer center "bureaucracy."

In fact, it is not only the case that the overall unit costs of computing are decreasing, but also that the economies of scale have changed radically. As Siewiorek, Bell, and Newell (1982, p. 333) note, the speed/cost ratio is 10 to 100 times *worse* for large high-speed processors than for current microprocessors. For example, Table 4 shows that for one major computer vendor, the speed/cost ratio of a recently introduced mainframe is about 30 times worse than the speed/cost ratio for a recent personal computer. Of course a simple comparison of processor speeds can easily be misleading since different machines have different memory sizes, different instruction sets, different operating systems, and so forth (e.g., see Siewiorek, Bell, and Newell, 1982). Nevertheless, there does appear to be a strong economic incentive for using small computers for as many tasks as possible.

In contrast to this advantage of small computers, our model also highlights a somewhat unexpected advantage of mainframes. Aside from factors such as storage capacity and software availability, our analysis suggests that the load leveling advantages of mainframes may be an important factor. Table 6 shows that (given our assumptions about arrival rate and service time distributions), the users of a mainframe would have to wait only 1/m times as long for their jobs as the users of m separate personal computers with the "equivalent" amount of total computational power. For equal cost configurations, the exact size and direction of this advantage depends on system load as well as on processor speeds and costs. Nevertheless, it is clear from our models that this factor provides a pressure toward mainframes.

In general, our model would predict that mainframes will be replaced for many future applications by one of the other organizational structures. Which structure is chosen should depend, in part, on the importance of communication costs and reliability: (1) In situations where communication costs are high, separate personal computers should be favored; (2) In situations where reliability is critical, networks of shared processors with decentralized scheduling should be most common; and

Table 4

Estimated Speed/Cost Ratios of Mainframe and Microcomputer from one Major Computer Vendor[1]

| *System* | IBM 3083-E | IBM Personal Computer |
|---|---|---|
| *Speed (MIPS)*[2] | 3.1 | .26 |
| *Purchase price*[3] | $1,200,000 | $3,353 |
| *Speed/cost ratio (MIPS/$M)*[4] | 2.58 | 77.6 |

Notes:

[1] These estimates are for rough comparison purposes only. Actual relative performance may vary with application, amount of memory, peripherals, software, and a number of other factors.

[2] Speed is measured in Million Instructions Per Second. Estimate for IBM 3083-E is by *Computerworld*, August 8, 1983, p. 31. Estimate for IBM Personal Computer is by David Bradley, Manger of Personal Systems Architecture for IBM Personal Computer (Personal communication with Amar Gupta, MIT Sloan School of Management, January 1984).

[3] Purchase price for processor only as of August, 1983. Source: *Computerworld*, August 8, 1983 and August 22, 1983.

[4] Million Instructions Per Second Per Million Dollars.

(3) In cases where neither communications costs nor reliability considerations are dominant, networks of shared processors with centralized scheduling should be preferred.

## Human organizations

### *Historical changes in business structures*

Table 5 summarizes, in simplified form, the changes in the dominant organizational structures used by American businesses described by Chandler (1966, 1977) and other business historians. From about 1850 to 1910, numerous small businesses coordinated by decentralized markets began to be superseded by large scale functionally organized hierarchies. These hierarchies continued to grow in size until, in the early and middle parts of this century, they were in turn replaced by the multi-divisional product hierarchies that are prevalent today. In the next section, we hypothesize how the widespread use of computers in organizations may again change the dominant organizational structures. Before doing that, however, our goal in this section is to show how these observed historical changes can be explained using the model we have presented. Williamson (1981b) and Chandler (1962, 1977) have also proposed explanations of these same changes and our explanation both draws on these earlier explanations and illuminates their incompleteness.

We assume that organizations move toward the structure that is best suited to their current situation. (For our purposes here, we do not care whether this motion results from conscious adaptation on the part of managers or from "blind" evolutionary forces favoring one kind of organization over another (see, e.g., Alchian, 1950; Nelson and Winter, 1981).) In our explanations, we will insist that, for each structural change, we be able to say what underlying parameters changed in the old structure and why this change caused the new structure to become the most desirable of the alternatives.

*Decentralized hierarchies to functional hierarchies.* The first change to be explained is the change from separate small companies to large scale functional hierarchies. Williamson (1981b) and Chandler (1977) both explain this change as the result of changing economies of scale so that large scale processors became much more economical than small ones. In our model, this means that the production cost advantage of the large scale forms over the others became even more pronounced than it already was from queuing considerations alone. (Note that this involves augmenting the simple model to include economies of scale in operating costs.) If this change is large enough, large scale organizations will supplant small ones up to the size where either the scale economies are exhausted or where further scale economies are counterbalanced by the increasing vulnerability of

Table 5

**Changes of Dominant Organizational Structures in American Businesses**

| *Approximate Dates* | *Structural Change* | |
|---|---|---|
| | *From* | *To* |
| 1850-1910 | Decentralized markets | Functional hierarchies |
| 1920-1960 | Functional hierarchies | Product hierarchies |

Sources: Summarized from Chandler (1966, 1977), Williamson (1981).

the large scale structure to component failure.

As Table 2 shows, our basic model is indifferent between a large scale centralized market and a large scale functional hierarchy. However, based on Williamson's (1981b) arguments for why transaction costs should be lower in a hierarchy (e.g., hierarchies are better able to deal with externalities and with uncertainties about highly specific assets), we should expect functional hierarchies to be preferred to centralized markets in many cases.

*Functional hierarchies to product hierarchies.* The next change to be explained is the change from functional hierarchies to product hierarchies. Williamson and Chandler explain this change, in part, by saying that as functional hierarchies grow larger their executive offices become increasingly overloaded by the demands of coordinating all the different projects across all the different functional departments. In a product hierarchy, the operational and tactical components of these coordination problems are delegated to the division managers, leaving the top executive officers free to concentrate on strategic questions.

This seems to be a plausible description of an advantage product hierarchies have over large functional hierarchies, but it leaves an essential question unanswered: Why did the functional hierarchies grow larger in the first place? Why didn't companies just grow until they exhausted the economies of scale and then let further demand be met by other companies of a similar size coordinated by a market? Williamson gives reasons for why hierarchies are sometimes superior to markets, but not for why they should become even better during the period in question.

Our model allows us to answer this question quite simply as follows (see Table 3): As decentralized markets grow in size, their coordination costs increase much more rapidly than the coordination costs for the equivalent functional hierarchies[4]. Thus, there will be situations where markets are preferred to functional hierarchies at one size, but where markets become less and less desirable as they grow because of increasing coordination costs. There is a pressure, then, for more and more of the activity that is coordinated by markets to be transferred into functional hierarchies in order to economize on coordination costs. This explains, then, why functional hierarchies continued to grow, as the marketplaces in which they operated grew, even after the underlying scale economies were exhausted.

We have still not explained, however, why these large functional hierarchies would change to product hierarchies. If functional hierarchies were superior to product hierarchies at the beginning of the period, why didn't they remain so at the end? Williamson's and Chandler's arguments rest

ERRATA: Corrected versions of Figures 1 & 2 are shown here.

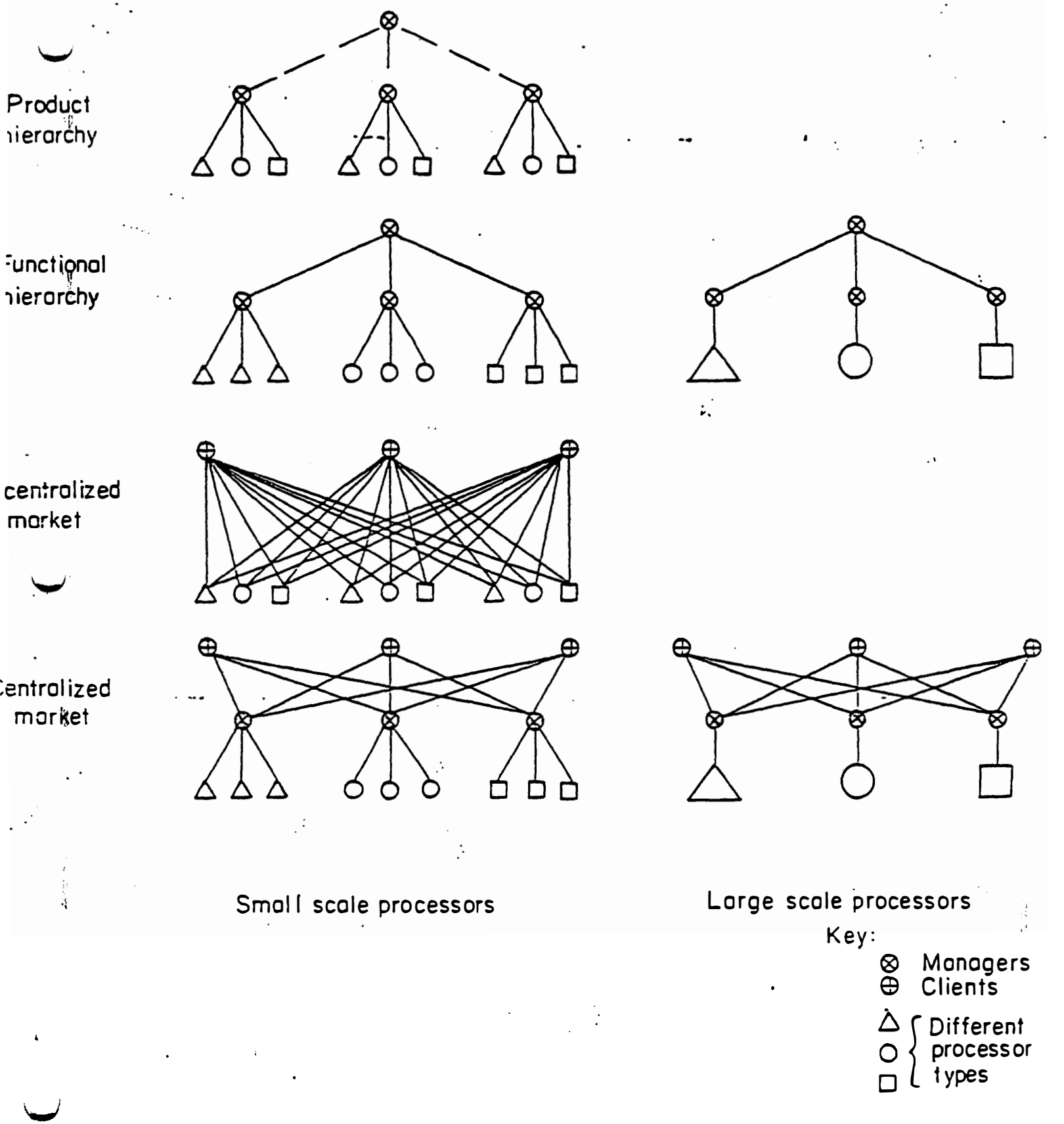
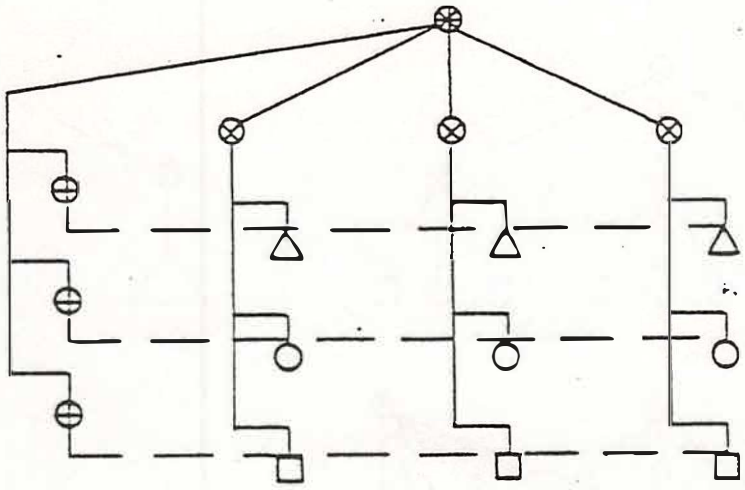



Figure 1
Alternative Organizational Forms



Key:

⊕ Functional manager
⊗ Product manager
⊛ Executive office

△ ○ □ Different processor types

Figure 2

Matrix Organization

on the assumption that the information processing capacity of a top management team is limited, no matter how many people are added to the team. If we don't make this assumption, however, Table 3 shows that there is no increasing advantage of functional hierarchies over product hierarchies as size increases.

There is an alternative explanation for the change, however, which is historically quite plausible. The argument is as follows: At the same time that functional hierarchies were getting larger, the relative importance of production costs and coordination costs was also changing. As production processes became more and more efficient, they constituted a smaller and smaller proportion of the total cost of products. Meanwhile, there were fewer improvements in the efficiency of coordination processes, so coordination costs constituted an increasing proportion of the total costs of products. Thus, product hierarchies, which economized on coordination costs at the expense of production costs, became increasingly attractive.

There is some strong empirical evidence to support this explanation. For example, Jonscher (1983) shows that the proportion of "information workers" in the workforce increased from about 25% in 1920 to almost 50% in 1960. During the same period, the economic productivity of "production workers" increased almost fourfold, while the productivity of information workers grew much more slowly. Jonscher also reports that during the period from 1947 to 1972 the proportion of total resources devoted to the two sectors (not just the number of workers) followed a similar trend. Taken together these results suggest that the relative importance of production and coordination costs did, indeed, change between 1920 and 1960, and that this might have contributed to the shift toward a less coordination-intensive organizational structure.

*Effect of widespread use of computers on organizational structure*

A number of authors have speculated about the structural changes in human organizations that are likely to result from the widespread use of computers (e.g., Strassman, 1980; Naismith, 1982; Toffler, 1970). In a few cases, observers have documented changes that have already resulted from the early uses of computers for data processing and management support (e.g., Robey, 1981, 1983; Walton & Vittori, 1983; Kling, 1980). Using either of these approaches as the basis for predicting long term trends is somewhat problematic, however. As Huber (in press) points out, these analysts may be extrapolating recent trends of a transition period far beyond the range where such extrapolation is valid. In particular, it is difficult to use the early effects of our first systems as the basis for predicting the ultimate effects of systems that, in some cases, have not even been developed yet.

In contrast to these approaches, the model we have presented here provides a principled basis for making long-term predictions based on very fundamental considerations. Though this approach is certainly not without its own dangers, the ability of our model to provide plausible explanations for the historical changes that led organizations to have the forms they do today gives us some additional confidence in its validity.

It seems plausible, first of all, to hypothesize that the widespread use of computers in organizations may substantially decrease the "unit costs" of coordination--both the transmission and processing of information. If this happens, then in some situations, coordination costs that would previously have been prohibitively high will become affordable. This could have at least two possible effects (see Table 2). The first possibility is that product hierarchies will shift toward functional hierarchies in order to take advantage of the accompanying reductions in production costs. We expect this to be the preferred adaptation in industries or companies where economizing on production costs is the most important strategic consideration.

For many industries and companies, however, we believe that retaining maximum flexibility will be an even more important consideration (e.g., see Piori, 1983; Piori & Sabel, in press; Huber, in press). In order to achieve this flexibility, our model predicts that they should shift toward being more like decentralized markets. The higher coordination requirements of market-like structures will be more affordable, and markets provide the additional flexibility of being less vulnerable to situational changes.

One possibility is that this change will involve a gradual replacement of large hierarchies with numerous small firms whose activities are coordinated by a computer-mediated decentralized market. The increasing importance of small entrepeneurial companies in many high technology markets--particularly computers--provides an early indication of this trend.

Another, and perhaps more likely, possibility is that the coordination mechanisms actually used inside large firms will come to resemble the structure of a decentralized market more than that of a rigid hierarchy. For example, the widespread use of electronic mail, computer conferencing, and electronic bulletin boards can facilitate what some observers (e.g., Mintzberg, 1979; Toffler, 1970) have called "adhocracies," that is, rapidly changing organizations with many shifting project teams composed of people with different skills and knowledge. Electronic media can help bring together the right people and skills for these teams. Matching skills to needs is only part of the problem, however. It is also necessary to provide incentives for people to perform tasks for many different rapidly changing project teams. One interesting possibility is that computer-mediated internal

markets may lower at least some of the transaction costs and thus enable much wider use of internal transfer payments to provide these incentives.

## CONCLUSIONS

In this paper, we have shown how viewing organizational structures as alternative solutions to the task assignment problem clarifies fundamental tradeoffs in organizational design. Though this model is a simplification of the immense complexity of real organizations, we were surprised to find what a wide variety of phenomena it helps to explain. It summarizes a number of traditional propositions about organizational design (e.g., the advantages and disadvantages of product hierarchies). The model also has implications for designing computer network schedulers, for explaining historical changes in American business structure, and for predicting future organizational changes that may result from the widespread use of computers.

Just as cognitive psychology and artificial intelligence appear to have benefited from simultaneously considering how human brains and computers might solve similar problems, we believe that computer science and organization theory both stand to gain a great deal from simultaneously considering how human organizations and computer systems might solve similar problems.

For example, it has become common in recent years for behavioral theorists to use information processing concepts to describe human organizations (e.g., March & Simon, 1958; Cyert & March, 1963; Galbraith, 1973, 1977; Hurwicz, 1973; Williamson, 1975; Cohen, 1982a, 1982b). There are also an increasing number of computer systems whose designers have made explicit use of metaphors from human organizations in structuring their systems (Goldberg & Robson, 1983; Hewitt, 1977; Erman et al, 1980; Smith, 1980; Smith & Davis, 1981; Kornfeld & Hewitt, 1981; Malone, Fikes, and Howard, 1983). So far, however, all these examples of cross-disciplinary fertilization have been at the level of analogies, with each field borrowing concepts and insights, but not strong principles, from the other. Our analysis of the task assignment problem, with its strong implications for both disciplines, is a contribution to a more fundamental approach in this emerging interdisciplinary field.

## Footnotes

[1] There may well be a great deal of ambiguity about what the goals of an organization really are (e.g., see Cyert and March, 1963, Ch.3). Different observers and different members of a given group may have different ideas about the goals of the group (e.g., to maximize profit or to provide employment), but one is not justified in calling a group an "organization" without at least some implicit notion of how to evaluate its success.

[2] We are indebted to Michael Cohen for helping to clarify this point.

[3] We have substituted "functional" and "product" for the terms used in the original: "process" and "purpose," respectively.

[4] In fact, in the simple form of the model presented here, coordination costs *per task* do not increase at all as the functional hierarchy increases in size.

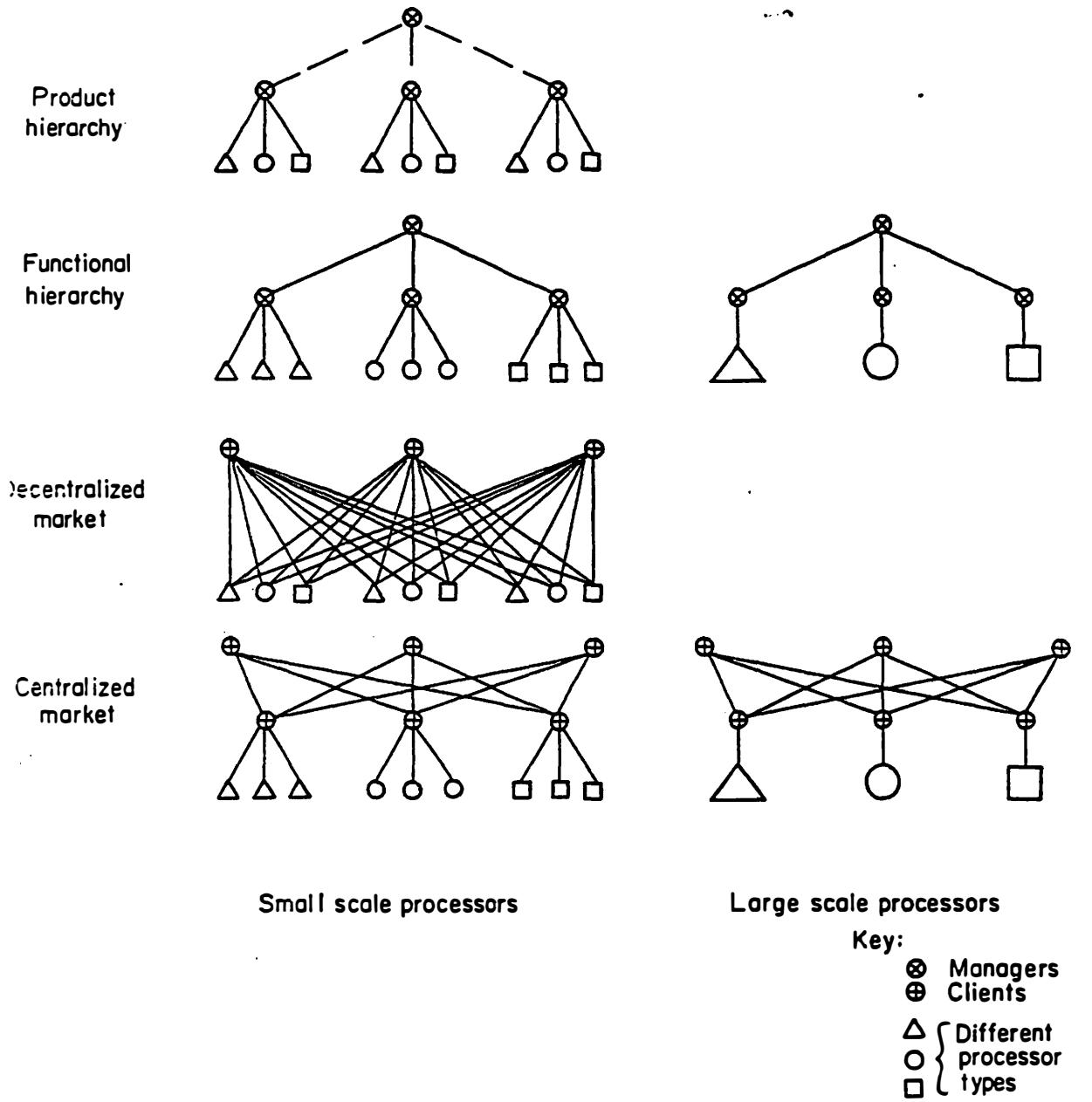
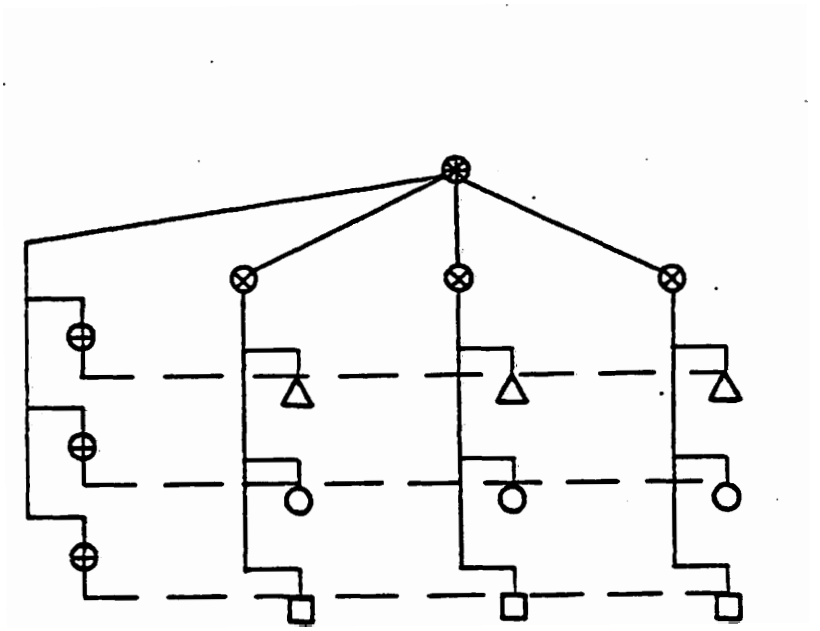



Figure 1
Alternative Organizational Forms



Key:

⊗ Functional manager
⊕ Product manager
⊛ Executive office

△ ○ □ Different processor types

Figure 2

Matrix Organization

## Acknowledgments

This research was supported by the Center for Information Systems Research at the Masssachusetts Institute of Technology, the Xerox Corporation Palo Alto Research Center, and National Science Foundation Grant No. SES-8213169.

The authors would like to thank Michael Cohen, Deborah Estrin, Amar Gupta, John Henderson, Charles Jonscher, Calvin Pava, Michael Scott Morton, Hoo-min Toong, Gordon Walker, and Joanne Yates for their helpful comments.

## References


Alchian, A. A. Uncertainty, evolution, and economic theory. *Journal of Political Economy*, 1950, *58*, 211-222.

Ansoff, H. I. & Brandenburg, R. G. A language for organization design (Parts I and II). *Management Science*, 1971, *17*, pp. B-705 to B-731.

Buzen, J. and Bondi, A. The response times of priority classes under preemptive resume in M/M/m queues. *Operations Research*, 1983, *31*, 456-466.

Chandler, A. D., *Strategy and Structure*, New York: Doubleday, 1962.

Chandler, A. D. *The visible hand: The managerial revolution in American business*. Cambridge, Mass.: Belknap Press, 1977.

Coase, R. H. The nature of the firm. *Economica N. S.*, 1937 (November), *4*, 386-405.

Cohen, M. D. *Conflict and complexity: Goal diversity and organizational search effectiveness*. Discussion paper #153 (revised), Institute of Public Policy Studies, University of Michigan, February 1982a.

Cohen, M. D. The power of parallel thinking. *Journal of Economic Behavior and Organization*, 1982b, in press.

Cyert, R. M. & March, J. G. *A behavioral theory of the firm*. Englewood Cliffs, N. J.: Prentice-Hall, 1963.

Davis, R., and Smith, R. G., *Negotiation as a Metaphor for Distributed Problem Solving* Artificial Intelligence Volume 20 Number 1, January 1983.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 1980, *12*, 213-253.

Farber, D. J. and Larson, K. C. The structure of the distributed computing system--Software. In J. Fox (Ed.), *Proceedings of the Symposium on Computer-Communications Networks and Teletraffic*, Brooklyn, NY: Polytechnic Press, 1972, pp. 539-545.

Fikes, Richard E., A Commitment-Based Framework for Describing Informal Cooperative Work, *Cognitive Science*, 1982, 6, 331-347.

Fox, M. S. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1981, *SMC-11*, 70-79.

Frazier, W. D. Potential technology implications for computers and telecommunications in the 1980's. *IBM Systems Journal*, 1979, *18*, 333-347.

Galbraith, J. *Designing complex organizations*. Reading, Mass.: Addison-Wesley, 1973.

Galbraith, J. *Organization design*. Reading, Mass.: Addison-Wesley, 1977.

Goldberg, A. & Robson, D. *Smalltalk-80: The language and its implementation*. Boston: Addison-Wesley, 1983.

Gulick, L. & Urwick, L. (Eds.) *Papers on the science of administration*. New York: Institute of Public Administration, Columbia University, 1937.

Hewitt, C. Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 1977, *8*, 323-364.

Huber, G. P. The nature and design of post-industrial organizations. *Management Science*, in press.

Hurwicz, L. The design of resource allocation mechanisms. *American Economic Review Papers and Proceedings*, 58 (May, 1973), 1-30 (Reprinted in K. J. Arrow and L. Hurwicz (Eds.) *Studies in Resource Allocation Processes*. Cambridge: Cambridge University Press, 1977, pp. 3-40).

Jones, A. K. & Schwarz, P. Experience using multiprocessor systems--A status report. *Computing Surveys*, 1980, *12*, 121-165.

Jonscher, C. *Information costs and efficiency under various organizational forms: A simulation model*. Unpublished paper, Department of Economics, Harvard University, November 1981 (Presented at the Transaction Costs Seminar, University of Pennsylvania, November 24, 1981).

Jonscher, C. Information resources and productivity. *Information Economics and Policy*, 1983, *1*, 13-35.

Kling, R. Social analyses of computing: Theoretical perspectives in recent empirical research. *Computing Surveys*, 1980, *12*, 61-110.

Kornfeld, W. A. & Hewitt, C. The scientific community metaphor. *IEEE Transactions on Systems, Man, and Cybernetics*, 1981, *SMC-11*, 24-33.

Lawrence, P. R., & Lorsch, J. W. *Organization and environment: Managing differentiation and integration*. Boston: Graduate School of Business Administration, Harvard University, 1967.

Lorin, H. Distributed processing: An assessment. *IBM Systems Journal*, 1979, *4*, 582-603.

Malone, T. W., Fikes, R. E., & Howard, M. T. *Enterprise: A market-like task scheduler for distributed computing environments.* Working paper, Cognitive and Instructional Sciences Group, Xerox Palo Alto Research Center, Palo Alto, Calif., October 1983.

March, J. G. & Simon, H. A. *Organizations.* New York: Wiley, 1958.

Mintzberg, H. *The structuring of organizations.* Englewood Cliffs, N. J.: Prentice-Hall, 1979.

Naisbitt, J. *Megatrends.* New York: Warner, 1982.

Nelson, R. & Winter, S. *An evolutionary theory of economic change.* Cambridge, Mass.: Harvard University Press, 1981.

Pfeffer, J. *Power in organizations.* Marshfield, Mass.: Pitman Publishing Co., 1981.

Piori, M. J. Computer technologies, market structure, and strategic union choices. Paper prepared for presentation to MIT/Union Conference on Industrial Relations in Transition, Cambridge, Mass., June 19, 1983.

Piori, M. J. & Sabel, C. *The second industrial divide.* New York: Basic Books, in press.

Robey, D. Computer Information Systems and Organization Structure. *Communications of the ACM*, 1981 (October), 679-687.

Robey, D. Information systems and organizational change: A comparative case study. *Systems, Objectives, Solutions*, 1983, *3*, 143-154.

Saaty, T. L. Resume' of useful formulas in queuing theory. *Operations Research*, 1957, *5*, (No. 2).

Scott, W. R. *Organizations: Rational, natural and open systems.* Englewood Cliffs, N. J.: Prentice-Hall, 1981.

Scott Morton, M. S. & Rockart, J. *Implications of changes in information technology for corporate strategy.* Working Paper No. 98, Center for Information Systems Research, Massachusetts Institute of Technology, January 1983.

Siewiorek, D. P., Bell, C. G., & Newell, A. *Computer Structures: Principles and Examples.* New York: McGraw-Hill, 1982.

Simon, H. A. *The sciences of the artificial* (Second Edition). Cambridge, Mass.: MIT Press, 1981.

Smith, R. G. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, December 1980, *C-29*, 1104-1113.

Smith, R. G., & Davis, R. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*. 1981, *SMC-11*, 61-69.

Strassman, P. S. The office of the future: Information management for the new age. *Technology Review*. December/January, 1980, pp. 54-65.

Thompson, J. D. *Organizations in action*. New York: McGraw-Hill, 1967.

Toffler, A. *Future Shock*. New York: Bantam Books, 1970.

Walton, R. E. & Vittori, W. New information technology: Organizational problem or opportunity? *Office: Technology and People*, 1983, *1*, 249-273.

Williamson, O. E. *Markets and hierarchies*. New York: Free Press, 1975.

Williamson, O. E. Transaction cost economics: The governance of contractual relations, *Journal of Law and Economics*, 1979, *22*, 233-261.

Williamson, O. E. The organization of work: A comparative institutional assessment. *Journal of Economic Behavior and Organization*, 1980, *1*, 6-38.

Williamson, O. E. The economics of organization: The transaction cost approach. *American Journal of Sociology*, 1981a, *87*, 548-575.

Williamson, O. E. The modern corporation: Origins, evolution, attributes. *Journal of Economic Literature*, 1981b, *XIX*, 1537-1568.

Woodward, J. *Industrial organization: Theory and practice*. New York: Oxford University Press, 1965.

# Appendix
## Formal justifications for organizational form comparisons

The bases for the qualitative comparisons of organizational forms in Tables 2 and 3 are summarized in Tables 6 and 7 and explained below. The following abbreviations are used: PH for product hierarchy, DM for decentralized market, CM for centralized market/functional hierarchy, and CM.S for centralized market/functional hierarchy (large scale).

For all organizational forms it is assumed that tasks of a given type arrive randomly according to a Poisson process with arrival rate $\lambda$ per "small" processor or $m\lambda$ per "large" processor. Individual tasks are processed at a rate $\mu$ on small processors and $m\mu$ on large processors. In some cases, processing times will be assumed to be exponentially distributed in order to obtain closed form expressions for the queue length statistics. This is usually a pessimistic assumption as far as performance is concerned, since the exponential has a mean to standard deviation ratio that is relatively high. When general service times are used, the variance of the service time will be denoted by $\sigma^2$.

The order of processing will be assumed to be on a first-come-first-served (FCFS) basis, subject to any priorities that may be assigned to the various tasks. In general, we will assume that the arriving tasks have many different possible priorities; in fact each task may have a unique priority. However, the probability distribution of processing time is assumed to be the same for all tasks. With this assumption, the average queue length and expected waiting time of an arbitrarily selected task are identical to those obtained when all tasks have the same priority. (If priority and processing time are not independent, the overall expected queue length may be larger or smaller than in the one priority case. For the single server queue, there are closed form expressions for the queue length statistics in the multi-priority, general service time case Saaty (1957). For the M/M/m queue with multiple priorities, there are closed form expressions for the queue length statistics for the preemptive resume service discipline (Buzen and Bondi, 1983). However, for this paper, the additional insight obtained from introducing these generalizations does not seem to justify the complexity they create.)

Based on the priorities of the tasks in queue, the order of the queue must be rearranged after each new arrival. Once the processing of a task begins, however, we assume that it can no longer be preempted by a higher priority task. The reordering of the queue is assumed to occur in parallel

**Table 6**

**Evaluation Criteria for Alternative Scheduling Organizations**

| *Organizational form* | *Expected Waiting Time* | *Messages per task* | *Expected Cost of Failure* |
|---|---|---|---|
| Product hierarchy | $1/(\mu - \lambda)$ | 2 | $mp_S c_S$ |
| Decentralized market | $(L_{DM}/m\lambda) + 1/\mu$ | $m(a+b) + 2$ | $mp_S c_D$ |
| Centralized market/ Functional hierarchy | $(L_{CM}/m\lambda) + 1/\mu$ | 4 | $mp_S c_D + p_{SCHED} c_L$ |
| Centralized market/ Functional hierarchy (Large scale) | $1/m(\mu - \lambda)$ | 4 | $p_L c_L + p_{SCHED} c_L$ |

Table 7

Rates of Change of Evaluation Criteria as Size of Economy Increases

| *Organizational form* | *Expected Waiting Time* | *Messages per task* | *Expected Cost of Failure* |
|---|---|---|---|
| Product hierarchy | 0 | 0 | $p_S c_S$ |
| Decentralized market | - | (a+b) | $p_S c_D$ |
| Centralized market/ Functional hierarchy | - | 0 | $p_S c_D + p_{SCHED} c_S$ |
| Centralized market/ Functional hierarchy (Large scale) | $-1/m^2(\mu - \lambda)$ | 0 | $(p_L + p_{SCHED}) c_S$ |

with the operation of the processor(s), thus requiring no interruption.

We assume that processors fail according to Poisson processes at constant rates. The rates are $p_S$ for small scale processors, $p_L$ for large scale processors, and $p_{SCHED}$ for central scheduling processors (with $p_L \geq p_S$, and $p_S$, $p_L$, $p_{SCHED} > 0$). The failure of one processor is assumed to be independent of the operational status of all other processors.

We assume that the expected cost of having the tasks on a small processor delayed (because the processor fails and they are sent elsewhere) is $c_D$. The expected cost of being unable to process tasks at all (because they cannot be sent elsewhere automatically) is assumed to be $c_S$ for the tasks on a small processor (with $c_S \geq c_D$) and $c_L$ for the tasks on a large processor (with $c_L \geq mc_S$ to include any additional cost of having the entire system fail at once).

We assume that there are m small processors and p product managers or clients. We will assume that when the economy increases in size, both the number of processors and the number of product managers increase at the the same rate (i.e., p/m is constant for all m).

**Product hierarchy**

In the product hierarchy organizational form, each processor operates autonomously. Thus the expected waiting time (including processing) for FCFS arrivals is given by the Pollaczek Khinchine Formula

$$W_{PH} = (\lambda/\mu^2)[1+\sigma^2\mu^2]/[2(1-\lambda/\mu)] + 1/\mu,$$

which for exponential service times reduces to

$$W_{PH} = 1/(\mu-\lambda).$$

In this form, there are m identical processors operating in parallel with no opportunity to divert tasks to other processors. Thus the expected failure cost per unit time is $mp_Sc_S$.

**Functional hierarchy/Centralized market**

In these two organizational forms, m processors of each kind are connected to a central scheduler. Each task arrives at the central scheduler, who directs the task to the processor that offers it the minimum delay to completion of service. When a task is completed, its result is returned, first to the scheduler and from there to the executive office or client. (If the result were sent directly from

the processor to the executive office or client, an extra message would be required anyway to notify the central scheduler that the processor had completed the task and was now available for other tasks.)

The arrival rate for the scheduler is $m\lambda$ and the processors themselves can be viewed as m servers for the queue. In this case, we will assume that the service times are exponentially distributed in order to obtain a closed form expression for the queue length and waiting time. This gives the expected queue length and waiting time, respectively, as follows:

$$L_{CM} = [(m\lambda/\mu)^m(\lambda/\mu)P_0]/[m!(1-\lambda/\mu)^2]$$

$$\text{and } W_{CM} = L_{CM}/m\lambda + 1/\mu,$$

$$\text{where } P_0 = 1/\left[\sum_{i=0,m-1} (m\lambda/\mu)^i/i! + [(m\lambda/\mu)^m/m!]/(1-\lambda/\mu)\right].$$

Since failure can arise from either processor failures or the scheduler failure, the expected failure cost per unit time is $mp_Sc_D+p_{SCHED}c_L$.

**Decentralized market**

In this case, tasks are generated by the clients who must select a procesor using some decentralized bidding scheme. We characterize the space of possible decentralized bidding schemes as follows: First, the client sends out $a$ announcements about the task per processor in the network. Thus if the announcement is broadcast to all processors, $a = 1$; if it is sent to only selected processors, $a < 1$; and if the task is announced to all processors more than once, $a > 1$. (This latter case occurs, for example, when tasks are "bumped" and rescheduled in the "eager" assignment method described by Malone, Fikes, and Howard (1983).) Then the client receives $b$ bids per processor, where $b$ may be greater than, less than, or equal to 1. In all cases, we assume that more than one processor is involved in the announcement and bidding cycle ($a, b > 1/m$).

If clients poll all contractors to find the best one (i.e., $a \geq 1$), then the expected aggregate queue length and waiting time for the DM case are equal to those of the CM case derived above. If not all clients are polled, then the waiting time for the DM case may be greater than for the CM case.

The expected failure cost per unit time is $mp_Sc_D$, since when a processor fails, its tasks are sent by their product managers to other processors.

Centralized market / Functional hierarchy (Large scale)

In this case, the number of messages sent is the same as for a centralized market / functional hierarchy (small scale). There is again a central scheduler who expedites the processing of the arriving tasks for the p product mangers. However, there is now a single processor with rate $m\mu$. Thus the queuing formulas for the M/G/1 queue apply with arrival rate $m\lambda$ and service rate $m\mu$. Substituting these rates into the formulas for the PH case, we obtain

$$L_{CMLS} = L_{PH}$$

$$W_{CMLS} = W_{PH}/m.$$

Since failure can result from either the processor failing or the scheduler failing, the expected failure cost per unit time is $p_L c_L + p_{SCHED} c_L$.

Comparisons

This section justifies the inequalities upon which the qualitative organizational form comparisons are based.

*Average waiting time*

It can be verified algebraically that, for $m > 1$: $W_{PH} > W_{CM} = W_{DM} > W_{CMLS}$. We first note that

$$W_{PH} = L_{PH}/\lambda + 1/\mu \quad = 1/(\mu - \lambda)$$

$$W_{CM} = L_{CM}/m\lambda + 1/\mu$$

$$W_{CMLS} = L_{PH}/m\lambda + 1/m\mu \quad = 1/[m(\mu - \lambda)].$$

The average queue length $L_{CM}$ is more simply expressed as

$$1/L_{CM} = [(1-\rho)^2/\rho] \Sigma_{i=0,m} a_i(m) + 1 - \rho \qquad \text{(A1)}$$

where $a_i(m) = [m/m][(m-1)/m][(m-2)/m] \ldots [(m-i+1)/m][1/\rho]^i$, and $\rho = \lambda/\mu$.

The form of the $a_i(m)$ results from dividing through by the numerator of $L_{CM}$ and then reversing the order of summation in the sum. Since $(m-k)/m < (m+1-k)/(m+1)$ for all $k > 0$, we see that $a_i(m+1) > a_i(m) \geq 0$ for all $i \geq 1$, and $a_0(m) = 1$. Thus $L_{CM}$ is strictly decreasing in m. Since,

for $m=1$, $L_{CM} = L_{PII}$, the decreasing property of $L_{CM}$ establishes $W_{PII} > W_{DM}$, for $m > 1$.

The second inequality, $W_{CM} > W_{CMLS}$ will hold if and only if

$$L_{CM}/m\lambda + 1/\mu > 1/[m(\mu - \lambda)]$$

or $$L_{CM} + m\rho > \rho/(1 - \rho)$$

or $$[(1 - \rho)/\rho]L_{CM} > 1 - m(1 - \rho)$$

or $$[1 - m(1 - \rho)]\rho/[L_{CM}(1 - \rho)] < 1.$$

From (A1), we see that

$$\rho/[L_{CM}(1 - \rho)] = (1 - \rho)\, s + \rho,$$

where $s = 1 + \Sigma_{i=1,m}\, a_i(m)$.

Thus we need to show that $[1 - m(1 - \rho)][(1 - \rho)\, s + \rho] < 1$.

If $1 - m(1 - \rho) < 0$, the result is obvious. Suppose, on the other hand, that $m(1-\rho) < 1$ or $\rho > (m-1)/m$. Then from (A1), we see that each of the terms $a_i(m)$, $i \geq 1$, satisfy $a_i(m) < 1/\rho$. Thus $s \leq 1 + m/\rho$. Therefore, it will be sufficient to show that $[1 - m(1 - \rho)][(1 - \rho)(1 + m/\rho) + \rho] < 1$. Multiplying this out and collecting terms, this reduces to $1 + m(1 - \rho)^2(1 - m)/\rho < 1$. This clearly holds for $m > 1$.

To see that $W_{DM} = W_{CM}$, we note that if, in the DM case, all contractors are polled to find the best one ($a \geq 1$), then the arrivals are processed in exactly the same way in these two systems. The only distinction is that in the DM case, tasks wait at their original arrival location, while in the CM case, tasks wait at the central server. (If not all processors are polled ($a < 1$), then there may be times when tasks in the DM case wait longer than in the CM case.)

It is interesting to note that this argument depends on the assumption of exponential service times. Sauer and Chandy (1979) show that for a slightly different problem (a two-stage cyclic queuing system with exponential delays between service requests), the expected waiting time for a large scale processor (CMLS) may actually be more than for a network of small processors (CM) if the coefficient of variation of service times is large enough. The reason for this result is that a few very long jobs can tie up the large scale processor completely for long periods of time, while in a network of small processors, a long job only ties up one of several processors.

*Coordination costs*

Given the minimum message requirements shown in Table 6 and the assumption that a, b > 1/m, the following inequalities for coordination costs, C, follow immediately: $C_{PH} < C_{CM} = C_{CMLS} < C_{DM}$.

*Expected cost of component failures*

Given the inequalities assumed about $p_S$, $p_L$, $p_{SCHED}$, $c_S$, $c_D$, and $c_L$ and the expressions for failure costs F in Table 6, the following inequalities all follow immediately: $F_{DM} < F_{PH}$, $F_{CM} < F_{CMLS}$. Note that the same results could have been obtained if, instead of assuming that $p_L > p_S$, and $c_L > mc_S$, we had assumed either of the following: (a) $p_L > mp_S$ and $c_L > c_S$, or (b) $p_L c_L > mp_S c_S$.

*Size of the economy*

In order to compare the rates of change of different criteria as the size of the economy changes, we will let $p = km$, and $c_L = mc_S + k'$, and examine the partial derivatives with respect to m. Table 5 shows these partial derivates except for the derivatives involving $W_{DM}$ and $W_{CM}$. It is clear that, as m grows, $W_{PH}$ remains constant, $W_{CM} = W_{DM}$ decreases asymptotically to $1/\mu$, and $W_{CMLS}$ decreases asymptotically to 0. The declines for all three of the latter values (which may be at different rates) are represented in Table 3 by a single minus sign. The assignment of varying numbers of pluses or minuses for the other values in Table 3 all follow immediately from the relative sizes of the partial derivatives in Table 7.

## Symbol Table

$a$ = number of announcements per processor in the economy, when an arriving task is put up for bids

$b$ = number of bids submitted per processor in the economy, when an arriving task is put up for bids

$c_D$ = expected task delay costs per small processor failure

$c_L$ = expected cost of a processing stoppage per large processor failure

$c_S$ = expected cost of a processing stoppage per small processor failure

$C_{PH}$, $C_{DM}$, $C_{CM}$, $C_{CMLS}$ = coordination cost per task for the various organizational forms

$d$ = additional delay per task required to wait for and process bids beyond delay required by a centralized scheduler

$D_{DM}$, $D_{CM}$ = expected total time lost per user per day for the various organizational forms

$f$ = time lost per user if the scheduler fails

$F_{PH}$, $F_{DM}$, $F_{CM}$, $F_{CMLS}$ = expected failure cost per unit time for the various organizational forms

FCFS = first-come-first-served queue discipline

$L_{PH}$, $L_{DM}$, $L_{CM}$, $L_{CMLS}$ = expected number of tasks in queue (excluding those in service) for the various organizational forms

$m$ = number of identical small scale processors

$p$ = number of product managers

$p_L$ = failure rate per unit time for large processors

$p_S$ = failure rate per unit time for small processors

$p_{SCHED}$ = failure rate per unit time for scheduling processors

$P_0$ = probability that there are no tasks in queue (excluding those in service)

SPTF = shortest-processing-time-first queue discipline

$t$ = expected number of tasks per user per day scheduled by the system

$W_{PH}$, $W_{DM}$, $W_{CM}$, $W_{CMLS}$ = expected waiting time per task (including service time) for the various organizational forms

$\lambda$ = arrival rate of tasks per small processor ($m\lambda$ = arrival rate per large processor)

$\mu$ = service rate of tasks per small processor ($m\mu$ = arrival rate per large processor)

$\sigma^2$ = variance of service time for tasks (when not constrained to have exponential service times)